Decision Trees

-- Decision Trees are probably the most popular and commonly used classification tasks.

-- Each case consists of a set of attributes with value and has a known class.

-- Classes are one of a small number of possible values, usually binary.

-- Attributes may be binary, multi-valued, or continuous.

-- Decision trees are recursively built following a top-down approach by repeated splits of the training set.

--A simple decision tree for classification of samples with two input attributes X_1 and X_2 is given in Figure 1.

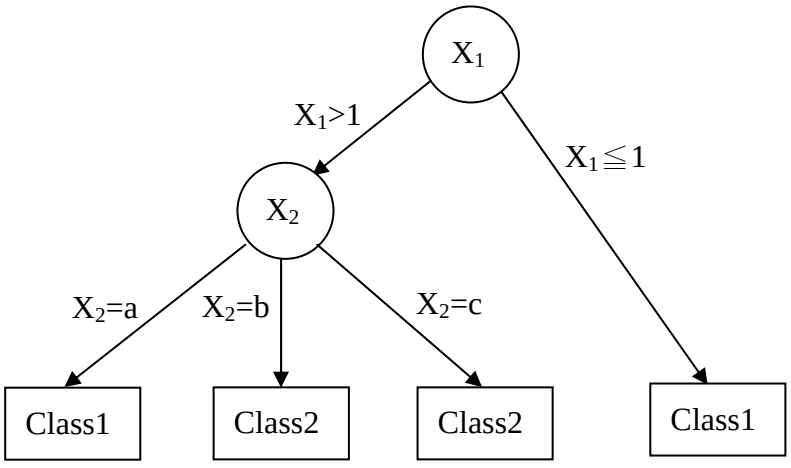


Figure 1 A simple decision tree with the tests on attributes X_1 and X_2

Generating a decision tree (Kantardzic, M., 2003)

--Decision trees are constructed beginning with the root of the tree proceeding down to its leaves.

-- Attributes selection in C4.5 algorithm is based on the information entropy measure applied to the examples at a node; the CART is base on the Gini's index.

C 4.5

Splitting criteria

-- When this dataset contains numerical attributes, binary splits are usually performed by choosing the threshold value which minimizes the impurity measure used as splitting criterion (Berzal, et al., 2004).

Information gain

$Gain = -p_i * \log_2 p_i$

where p_i is the number of training set that belong to class i.

-- While we have already explained standard test for categorical attributes, additional explanations are necessary about a procedure for establishing tests on attributes with numeric value. It might seem that tests on continuous attributes would be difficult to formulate, since they contain an arbitrary threshold for splitting all values into two intervals.

-- Any threshold value lying between v_i and v_{i+1} will have the same effects as dividing the cases into those whose value of the attribute Y lies in $\{v_1, v_2, ..., v_i\}$ and those whose value is in $\{v_i, v_{i+1}, ..., v_m\}$. There are thus only m-1 possible splits on Y, all of which should be examined systematically to obtain an optimal split.

-- It is usual to choose the midpoint of each interval, $(v_i + v_{i+1})/2$, as the representative threshold. The algorithm C4.5 differs in choosing as the threshold a smaller value v_i for every interval $\{v_i, v_{i+1}\}$, rather the midpoint itself.

A simple example [Kantardzic, M. 2003]

Table1 A simple flat databases

Objects	Attribute 1	Attribute 2	Attribute 3	Class
1	A	70	True	C1
2	A	90	True	C2
3	A	85	False	C2
4	A	95	False	C2
5	A	70	False	C1
6	B	90	True	C1
7	B	78	False	C1
8	B	65	True	C1
9	B	75	False	C1
10	C	80	True	C2
11	C	70	True	C2
12	C	80	False	C1
13	C	80	False	C1
14	C	96	False	C1

Information gain

$$Info(S) = -[\tfrac{9}{14} * \log_2(\tfrac{9}{14}) + \tfrac{5}{14} * \log_2(\tfrac{5}{14})] = 0.940$$

$$\begin{aligned} Info(A1) &= \tfrac{5}{14} * (-\tfrac{2}{5} * \log_2 \tfrac{2}{5} - \tfrac{3}{5} * \log_2 \tfrac{3}{5}) + \tfrac{4}{14} * (-\tfrac{4}{4} * \log_2 \tfrac{4}{4} \\ &\quad - \tfrac{0}{4} * \log_2 \tfrac{0}{4}) + \tfrac{5}{14} * (-\tfrac{3}{5} \log_2 \tfrac{3}{5} - \tfrac{2}{5} \log_2 \tfrac{2}{5}) \\ &= 0.694 \end{aligned}$$

$$Gain(A1) = Info() - Info(A1)$$
$$= 0.940 - 0.694$$
$$= 0.246$$

$$Info(A3) = \frac{6}{14} * (-\frac{3}{6} * \log_2 \frac{3}{6} - \frac{3}{6} * \log_2 \frac{3}{6}) + \frac{8}{14} * (-\frac{6}{8} * \log_2 \frac{6}{8} - \frac{2}{8} * / \log_2 \frac{2}{8})$$
$$= 0.892$$

$$Gain(A3) = 0.940 - 0.892 = 0.048$$

After a sorting process, the set of value for attribute 2 is {65, 70,75,78,80,85,90,95 96}, and the set of potential threshold value T is {65,70,75,78,80,85,90}. Out of these eight values the optimal T (with the highest information gain) should be selected.

$$Info(65) = \frac{1}{14} * (-\frac{1}{1} * \log_2 \frac{1}{1}) + \frac{13}{14} * (-\frac{8}{13} * \log_2 \frac{8}{13} - \frac{5}{13} * \log_2 \frac{5}{13})$$
$$= 0.893$$

$$Info(70) = \frac{4}{14} * (-\frac{3}{4} * \log_2 \frac{3}{4} - \frac{1}{4} * \log_2 \frac{1}{4}) + \frac{10}{14} * (-\frac{6}{10} * \log_2 \frac{6}{10} - \frac{4}{10} * \log_2 \frac{4}{10})$$
$$= 0.925$$

$$Info(75) = \frac{5}{14} * (-\frac{4}{5} * \log_2 \frac{4}{5} - \frac{1}{5} * \log_2 \frac{1}{5}) + \frac{9}{14} * (-\frac{5}{9} * \log_2 \frac{5}{9} - \frac{4}{9} * \log \frac{4}{9})$$
$$= 0.895$$

$$Info(78) = \frac{6}{14} * (-\frac{5}{6} * \log_2 \frac{5}{6} - \frac{1}{6} * \log_2 \frac{1}{6}) + \frac{8}{14} * (-\frac{4}{8} * \log_2 \frac{4}{8} - \frac{4}{8} * \log_2 \frac{4}{8})$$
$$= 0.850$$

$$Info(80) = \frac{9}{14} * (-\frac{7}{9} * \log_2 \frac{7}{9} - \frac{2}{9} * \log_2 \frac{2}{9}) + \frac{5}{14} * (-\frac{2}{5} * \log_2 \frac{2}{5} - \frac{3}{5} * \log_2 \frac{3}{5})$$
$$= 0.838$$

$$Info(85) = \frac{10}{14} * (-\frac{7}{10} * \log_2 \frac{7}{10} - \frac{3}{10} * \log_2 \frac{3}{10}) + \frac{4}{14} * (-\frac{2}{4} * \log_2 \frac{2}{4} - \frac{2}{4} * \log_2 \frac{2}{4})$$
$$= 0.915$$

$$Info(90) = \frac{12}{14} * (-\frac{8}{12} * \log_2 \frac{8}{12} - \frac{4}{12} * \log_2 \frac{4}{12}) + \frac{2}{14} * (-\frac{1}{2} * \log_2 \frac{1}{2} - \frac{1}{2} * \log_2 \frac{1}{2})$$
$$= 0.930$$

$$Info(95) = \frac{13}{14} * (-\frac{8}{13} * \log_2 \frac{8}{13} - \frac{5}{13} * \log_2 \frac{5}{13}) + \frac{1}{14} * (-\frac{1}{1} * \log_2 \frac{1}{1})$$
$$= 0.893$$

Thus, $Info(A2) = Info(80)$; $Gain(A2) = 0.940 - 0.838 = 0.102$

Since A1 has the highest information gain of 0.246 and therefore this attributes will be

selected for the first splitting in the construction of a decision tree.

The root node will have the test for the values of attribute 1, and three branches will be created, one for each of the attribute values. The initial tree with the corresponding subsets of objects in the children nodes is represented in figure 2.

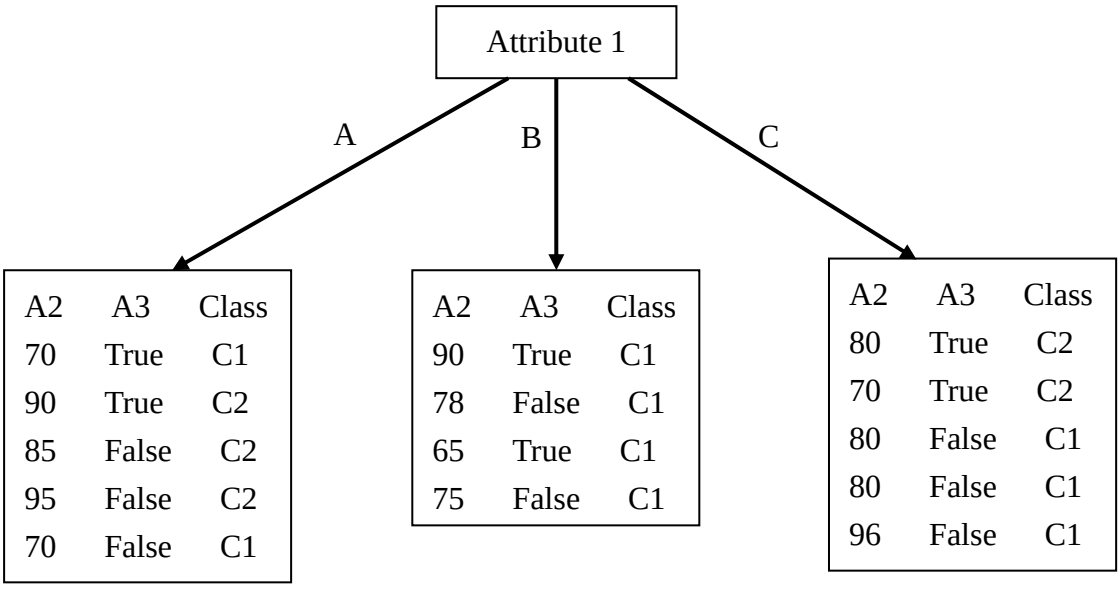


Figure 2 the initial decision tree

After initial splitting, each child node has several objects from database, and the entire process of test selection and optimization will be repeated for each child node.

Attribute=A

$$Info(S_1) = -\frac{2}{5} * \log_2 \frac{2}{5} - \frac{3}{5} * \log_2 \frac{3}{5} = 0.971$$

$$Info(A3) = \frac{2}{5} * (-\frac{1}{2} * \log_2 \frac{1}{2} - \frac{1}{2} * \log_2 \frac{1}{2}) + \frac{3}{5} * (-\frac{1}{3} * \log_2 \frac{1}{3} - \frac{2}{3} * \log_2 \frac{2}{3}) = 0.951$$

$$Gain(A3) = 0.971 - 0.951 = 0.020$$

After the initial splitting, the set of value for attribute 2 under attribute 1 value equal to A is {70, 85, 90,95}, and the set of potential threshold value *T* is {70,85,90}.

$$Info(70) = \frac{2}{5} * (-\frac{2}{2} * \log_2 \frac{2}{2}) + \frac{3}{5} * (-\frac{3}{3} * \log_2 \frac{3}{3}) = 0$$

$$Info(85) = \frac{3}{5} * (-\frac{2}{3} * \log_2 \frac{2}{3} - \frac{1}{3} * \log_2 \frac{1}{3}) + \frac{2}{5} * (-\frac{2}{2} * \log_2 \frac{2}{2}) = 0.551$$

$$Info(90) = \frac{4}{5} * (-\frac{2}{4} * \log_2 \frac{2}{4} - \frac{2}{4} * \log_2 \frac{2}{4}) + \frac{1}{5} * (-\frac{1}{1} * \log_2 \frac{1}{1}) = 0.800$$

Thus, $Info(A2) = Info(70) = 0$; $Gain(A2) = 0.971 - 0 = 0.971$

Since A2 has the highest information gain of 0.971 and therefore this attributes will be selected for the second child of the root node. An optimal threshold value is 70, and the branches of tree are Attribute 2$\leqq$70 and Attribute 2>70. A similar computation will be carried out for the third child of the root node. The final decision tree is shown in figure 3.

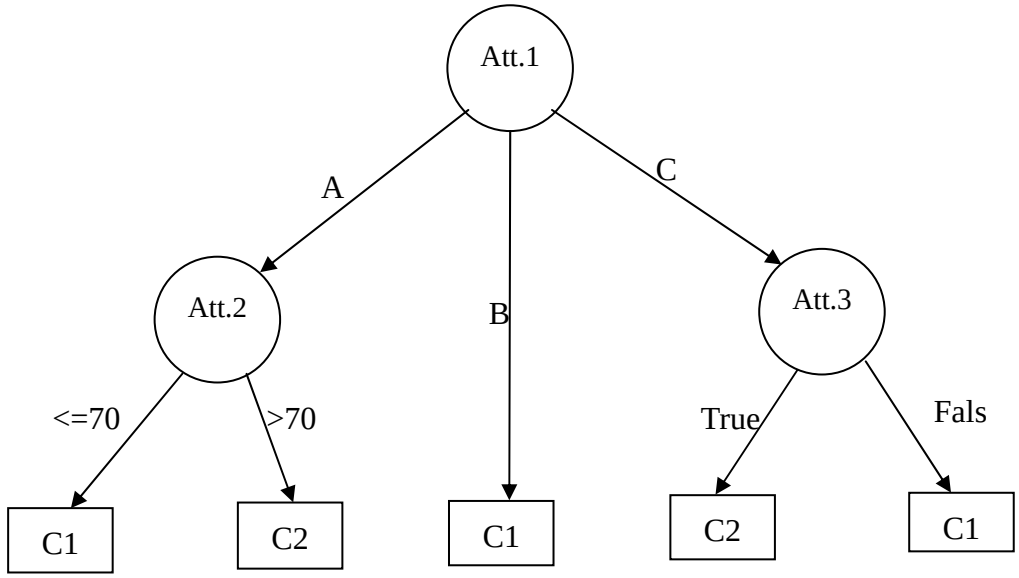


Figure 3 the final decision tree for C4.5 algorithm

Extracting rules

1. If $A_1 = A$ and $A_2 \leq 70$, then C_1. (2/0)
2. If $A_1 = A$ and $A_2 > 70$, then C_2. (3/0)
3. If $A_1 = B$, then C_1. (4/0)
4. If $A_1 = C$ and $A_3 = True$, then C_2. (2/0)
5. If $A_1 = C$ and $A_3 = False$, then C_1. (3/0)

Information gain ration **[Kantardzic, M., 2003]**

-- Attributes with multiple values favored by information gain. A solution was found in some kinds of normalization. By analogous with the definition of $Info(X)$, an additional parameter was specified:

$$Split-\inf o(X)=-\sum_{i=1}^{n}\frac{|T_i|}{|T|}*\log_2\frac{|T_i|}{|T|}$$

$$Gain-ratio(X)=\frac{Gain(X)}{Split-\inf o(X)}$$

For example, find the gain-ratio measure for the attribute 1.

$$Split-\inf o(A1)=-(5/14*\log_2 5/14+4/14*\log_2 4/14+5/14*\log_2 5/14)$$
$$=1.577$$

$$Gain-ratio(A1)=\frac{0.246}{1.577}=0.156$$

-- A similar procedure should be performed for other attributes in the decision tree. Instead of gain measure, the maximal gain ratio will be the criterion for attribute selection, along with a threshold to split objects into subsets.

Missing Attribute Values

--In a data set, some attribute values for some objects can be missing. Such incompleteness is typical in real world applications. To solve the problem of missing values, there are two choices:

1. Discard all objects with missing data.
2. Define a new algorithm or modify an existing algorithm.

In C4.5, to handle the objects with missing values are distributed probabilistically according to the relative frequency of known values. The new gain criterion will have the form

$$Gain(X) = F * (Info(S) - Info(X))$$

where

$$F = \frac{number\ of\ objects\ in\ the\ data\ set\ with\ a\ known\ value\ for\ a\ given\ attribute}{total\ number\ of\ objects\ in\ a\ data\ set}$$

--When an object with known value is assigned from *S* to S_i, the probability to T_i is 1, and in all subsets is 0. When a value is unknown, only a weaker probabilistic statement can be made.

--In C4.5, each object (having a missing value) in subset S_i was assigned a weight *w*, representing the probability that the object belongs to each subset.

$$w^{new} = w^{old} * P(S_i)$$

where $P(S_i)$ = the probability that object belongs to each subset.

For example

Table 2 A Simple flat databases with missing values

Objects	Attribute 1	Attribute 2	Attribute 3	Class
1	A	70	True	C1
2	A	90	True	C2
3	A	85	False	C2
4	A	95	False	C2
5	A	70	False	C1
6	**?**	90	True	C1
7	B	78	False	C1
8	B	65	True	C1
9	B	75	False	C1
10	C	80	True	C2
11	C	70	True	C2
12	C	80	False	C1
13	C	80	False	C1
14	C	96	False	C1

$$Info(S) = -(8/13 * \log_2 8/13 + 5/13 * \log_2 5/13) = 0.961$$

$$Info(A1) = 5/13 * (-2/5 * \log_2 2/5 - 3/5 * \log_2 3/5) + 3/13 * (-3/3 * \log_2 3/3) + 5/13 * (-3/5 * \log_2 3/5 - 2/5 * \log_2 2/5) = 0.747$$

$$Gain(A1) = 13/14 * (0.961 - 0.747) = 0.199$$

$$Split - \inf o(A1) = -(5/13 * \log_2 5/13 + 3/13 * \log_2 3/13 + 5/13 * \log_2 5/13) = 1.876$$

$$Gain - ratio(A1) = \frac{0.199}{1.876} = 0.106$$

— After splitting the set S into subsets using Attribute 1, the recode with the missing value will be represented in all three subsets. The results are given in Figure 4. New weight w_i will be equal to probability $5/13, 3/13,$ and $5/13$. The new values computed are $|T_1| = 5 + 5/13$, $|T_2| = 3 + 3/13$, and $|T_3| = 5 + 5/13$.

— If these subsets are partitioned further by the tests on Attribute2 and Attribute3, the final decision tree for a data set with missing values has the form shown in Figure 5.

— Every decision is attached with two parameters in the form ($|T_i| / E$).

Where $|T_i|$ is the sum of the fractional training samples reach the leaf;

E is the number of samples that belong to classes other than the nominated class.

For example,

(3.4/0.4) means that 3.4 (3+5/13) fractional training samples reach the leaf, which 0.4 (5/13) did not belong to the class assigned to the leaf. It is possible to

express the $|T_i|$ and E parameters in percentages:

$\frac{3}{3.4} * 100\% = 88\%$ of cases at a given leaf would be classified as C2.

$\frac{0.4}{3.4} * 100\% = 12\%$ of cases at a given leaf would be classified as C1.

S_1: (Attribute 1=A)

Attribute 2	Attribute 3	Class	Weight
70	True	1	1
90	True	2	1
85	False	2	1
95	False	2	1
70	False	1	1
90	**True**	**1**	**5/13**

S_2: (Attribute 1=B)

Attribute 2	Attribute 3	Class	Weight
90	**True**	**1**	**3/13**
78	False	1	1
65	True	1	1
75	False	1	1

S_3: (Attribute 1=C)

Attribute 2	Attribute 3	Class	Weight
80	True	2	1
70	True	2	1
80	False	1	1
80	False	1	1
96	False	1	1
90	**True**	**1**	**5/13**

Figure 4 Results of Attribute1 are subsets S_i (initial S with missing value)

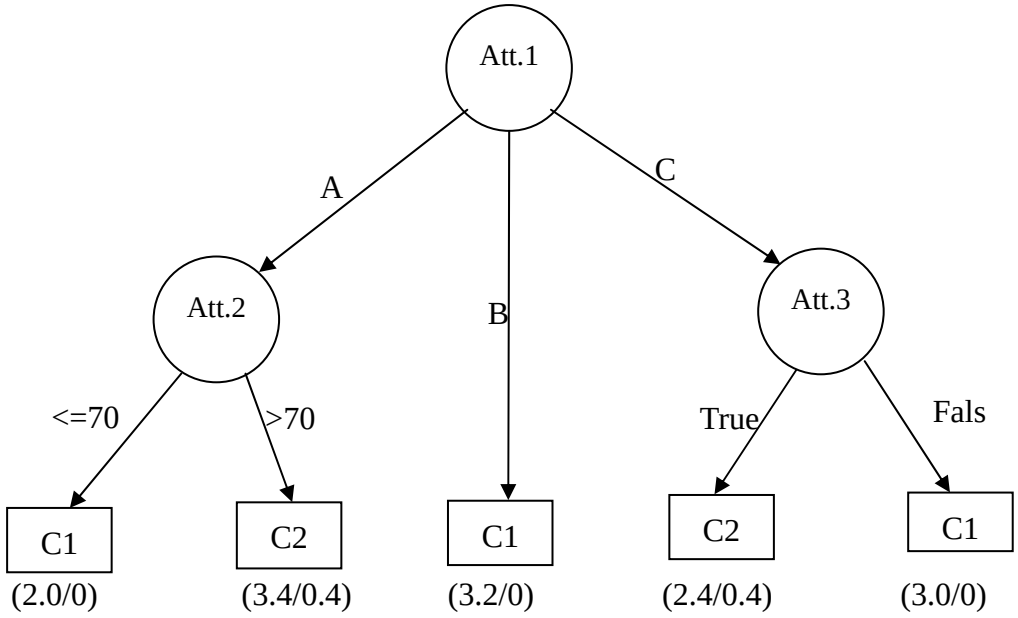


Figure 5 Decision tree for data set S with missing values

Pruning Decision Trees

—Discarding one or more sub-trees and replacing them with leaves simplify a decision tree, and that is the main task in decision tree pruning.

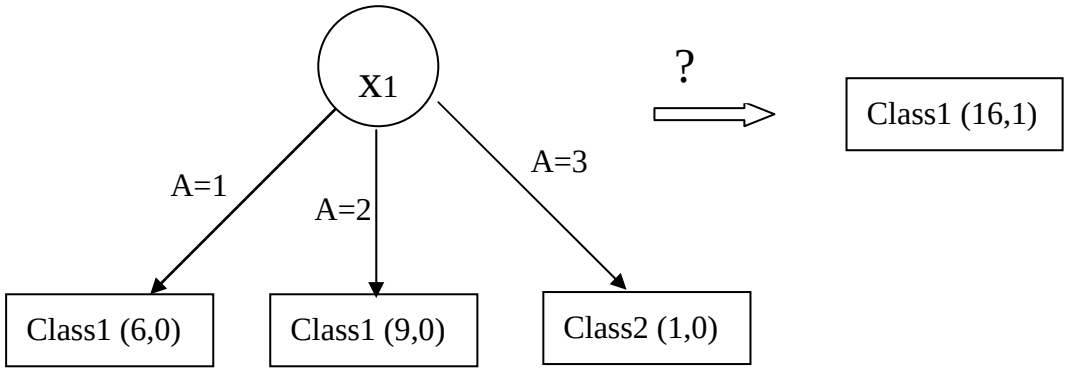
—The basic idea of decision tree pruning is to remove parts of the tree (sub-tree) that do not contribute to the classification accuracy of unseen testing objects, producing a less complex and thus more comprehensible tree.

—For every node in a tree, the estimation of the upper confidence limit U_{cf} is computed using the statistical tables for binomial distribution. Parameter U_{cf} is a function of $|T_i|$ and E for a given node. C4.5 uses the default confidence level of 25%, and compares $U_{25\%} * (\frac{|T_i|}{E})$ for a given node T_i with a weighted

confidence of its leaves. Weights are the total number of cases for every leaf.

If the predicted error for a root node in a sub-tree (PE_{node}) is less than weighted sum of $U_{25\%}$ for the leaves (PE_{tree}), then a sub-tree will be replaced with its root node, which becomes a new leaf in a pruned tree.

For example



Using default confidence of 25%, the upper confidence limits for all nodes are collected from statistical tables:

$U_{25\%}(6,0) = 0.178,\ U_{25\%}(9,0) = 0.075,\ U_{25\%}(1,0) = 0.75,\ U_{25\%}(16,1) = 0.063$

Using these values, the predicted errors for the initial tree and the replaced node are:

$$PE_{tree} = (6 * 0.178) + (9 * 0.075) + (1 * 0.75) = 2.493$$
$$PE_{node} = 16 * 0.063 = 1.008$$

Since $PE_{node} < PE_{tree}$, then the sub-tree be pruned and the sub-tree replaced with the new leaf node.

CART (Classification and Regression Trees)

— CART, an acronym for Classification And Regression Trees, is described in the book by Breiman et al., (1984)

— CART constructs trees that have only binary splits.

Split criterion

— Definition:

$$Gini(S) = 1 - \sum p_j^{\,2}$$

where S is a dataset containing examples from *n* classes.

p_j is a relative frequency of class *j* in *S*.

E.g., two classes, *Pos* and *Neg* , and dataset *S* with *p* *Pos* samples and *n* *Neg* samples.

$$P_{Pos} = p \Big/ p+n \qquad P_{Neg} = n \Big/ p+n$$

$$Gini(\mathrm{S}) = 1 - P_{Pos}^{\,2} - P_{Neg}^{\,2}$$

— If dataset S is split into S_1 and S_2, then splitting index is defined as follows:

$$G_{split}(S) = \frac{p_1 + n_1}{p+n} * Gini(S_1) + \frac{p_2 + n_2}{p+n} * Gini(S_2)$$

where p_i and n_i denote p_i *Pos* samples and n_i *Neg* samples in the dataset S_i.

— In this definition, the “best” split point is the midpoint of the lowest value interval $[v_i, v_{i+1}]$.

A simple car insurance example

Objects	Age	Car type	Risk
1	23	Family	High
2	17	Sport	High
3	43	Sport	High
4	68	Family	Low
5	32	Truck	Low
6	20	Family	High

After a sorting process, the possible values for age attribute are: $age \le 17$, $age \le 20$, $age \le 23$, $age \le 32$, $age \le 43$ and $age \le 68$.

$Gini\,(age \le 17) = 1 - (1^2 + 0^2) = 0$;

$Gini\,(age > 17) = 1 - [(3/5)^2 + (2/5)^2] = \frac{12}{25}$

$G_{split} = \frac{1}{6} * 0 + \frac{5}{6} * \frac{12}{25} = 0.4$

$Gini\,(age \le 20) = 1 - [(2/2)^2 + 0^2] = 0$

$Gini\,(age > 20) = 1 - [(2/4)^2 + (2/4)^2] = \frac{1}{2}$

$G_{split} = \frac{2}{6} * 0 + \frac{4}{6} * \frac{1}{2} = 0.333$

$Gini\,(age \le 23) = 1 - [(3/3)^2 + (0/3)^2] = 0$

$Gini\,(age > 23) = 1 - [(1/3)^2 + (2/3)^2] = \frac{4}{9}$

$G_{split} = \frac{3}{6} * 0 + \frac{3}{6} * \frac{4}{9} = 0.222$

$Gini\,(age \le 32) = 1 - [(3/4)^2 + (1/4)^2] = \frac{3}{8}$

$Gini\,(age > 32) = 1 - [(1/2)^2 + (1/2)^2] = \frac{1}{2}$

$G_{split} = \frac{4}{6} * \frac{3}{8} + \frac{2}{6} * \frac{1}{2} = \frac{5}{12} = 0.416$

$Gini\,(age \le 43) = 1 - [\{4/5)^2 + (1/5)^2] = \frac{8}{25}$

$$Gini\,(age > 43) = 1 - [(0/1)^2 + (1/1)^2] = 0$$

$$G_{split} = \frac{5}{6} * \frac{8}{25} + \frac{1}{6} * 0 = \frac{4}{15} = 0.267$$

$$Gini\,(age \le 68) = 1 - [(4/6)^2 + (2/6)^2] = \frac{4}{9}$$

$$Gini\,(age > 68) = 1 - [0^2 + 0^2] = 1$$

$$G_{split} = \frac{6}{6} * \frac{4}{9} + 0 * 1 = \frac{4}{9} = 0.444$$

The lowest value of G_{split} is for $age \le 23$, thus the split point for attribute age is:

(23+32)/2=27.5

$$Gini\,(Car\ type = Family) = 1 - [(2/3)^2 + (1/3)^2] = \frac{4}{9}$$

$$Gini\,(Car\ type \ne Family) = 1 - [(2/3)^2 + (1/3)^2] = \frac{4}{9}$$

$$G_{split} = \frac{3}{6} * \frac{4}{9} + \frac{3}{6} * \frac{4}{9} = \frac{4}{9} = 0.444$$

$$Gini\,(Car\ type = Sport) = 1 - [(2/2)^2 + (0/2)^2] = 0$$

$$Gini\,(Car\ type \ne Sport) = 1 - [(2/4)^2 + (2/4)^2] = \frac{1}{2}$$

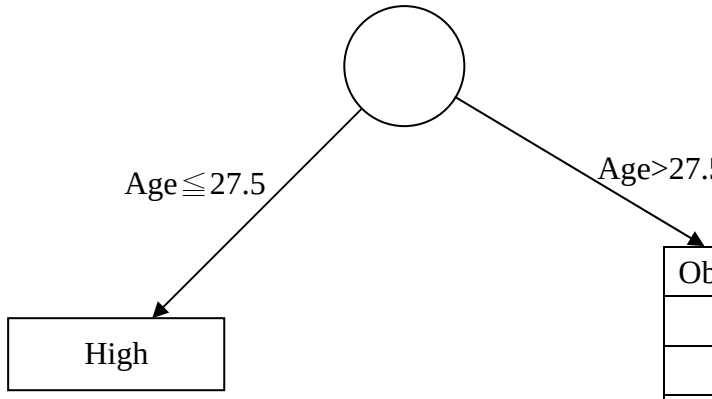
$$G_{split} = \frac{2}{6} * 0 + \frac{4}{6} * \frac{1}{2} = \frac{1}{3} = 0.333$$

$$Gini\,(Car\ type = Truck) = 1 - [(0/1)^2 + (1/1)^2] = 0$$

$$Gini\,(Car\ type \ne Truck) = 1 - [(4/5)^2 + (1/5)^2] = \frac{8}{25}$$

$$G_{split} = \frac{1}{6} * 0 + \frac{5}{6} * \frac{8}{25} = \frac{4}{15} = 0.267$$

After the first split, the decision tree of the example set is:



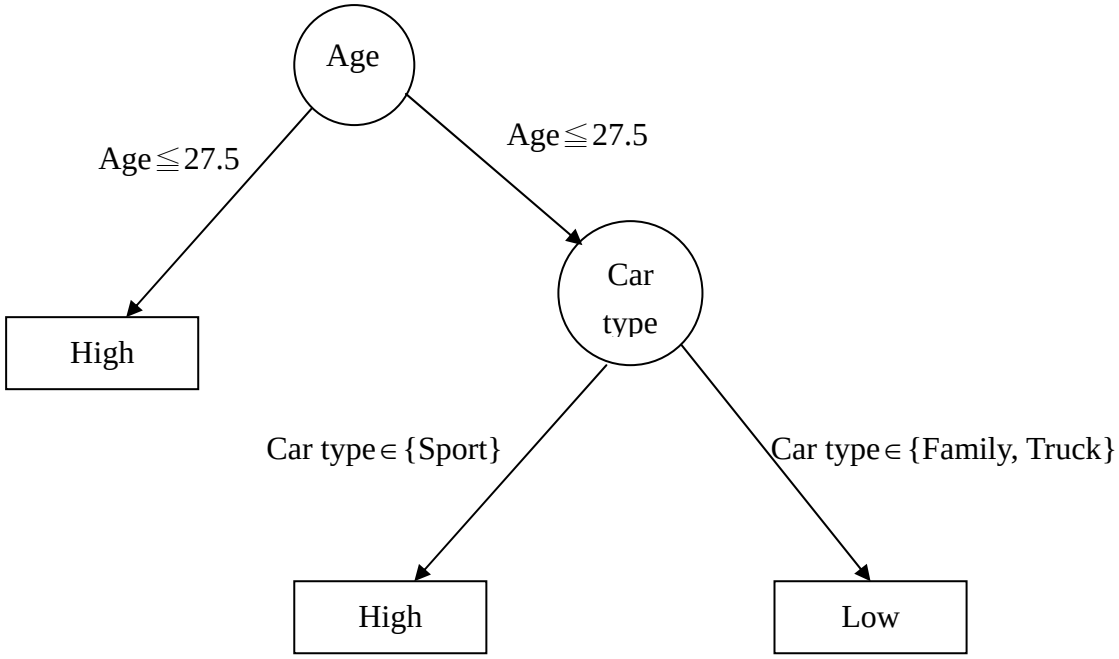
Objects	Car type	Risk
3	Sport	High
4	Family	Low
5	Truck	Low

$$Gini\,(Car\ type = Sport) = 1 - [(1/1)^2 + (0/1)^2] = 0$$
$$Gini\,(Car\ type \neq Sport) = 1 - [(0/2)^2 + (2/2)^2] = 0$$
$$G_{split} = \frac{1}{3} * 0 + \frac{2}{3} * 0 = 0$$

$$Gini\,(Car\ type = Family) = 1 - [(0/1)^2 + (1/1)^2] = 0$$
$$Gini\,(Car\ type \neq Family) = 1 - [(1/2)^2 + (1/2)^2] = \frac{1}{2}$$
$$G_{split} = \frac{1}{3} * 0 + \frac{2}{3} * \frac{1}{2} = \frac{1}{3} = 0.333$$

$$Gini\,(Car\ type = Truck) = 1 - [(0/1)^2 + (1/1)^2] = 0$$
$$Gini\,(Car\ type \neq Truck) = 1 - [(1/2)^2 + (1/2)^2] = \frac{1}{2}$$
$$G_{split} = \frac{1}{3} * 0 + \frac{2}{3} * \frac{1}{2} = \frac{1}{3} = 0.333$$

The lowest value of G_{split} is for $Car\ type = Sport$, the decision tree after the second split of the example set is:



Extracting rules

1. If $Age \le 27.5$, then $Risk = High$. (3/3)
2. If $Age > 27.5$ and $Car\ type \in \{Sport\}$, then $Risk = High$. (1/1)
3. If $Age > 27.5$ and $Car\ type \in \{Family, Truck\}$, then $Risk = Low$. (2/2)